



## COURS DE LANGAGE PASCAL

### SÉRIE 01

#### OBJECTIF PÉDAGOGIQUE :

À la fin de cette série, le stagiaire doit être capable d'acquérir les éléments de base d'un langage pascal.

#### PLAN DE LA LEÇON :

#### **I- INTRODUCTION AU LANGAGE PASCAL**

- 1- Caractéristiques générales du langage pascal
- 2- Environnement technique du langage pascal
- 3- Notations fondamentales

#### **II- ÉLÉMENTS DE BASE**

- 1- Alphabet pascal
- 2- Éléments d'un texte pascal
  - 2.1- Les mots clés pascal
  - 2.2- Les identificateurs
  - 2.3- Les séparateurs
  - 2.4- Les operateurs
- 3- Les commentaires
- 4- Structure générale d'un programme pascal

#### **RÉSUMÉ**

#### **EXERCICES D'APPLICATION**

#### **CORRIGÉ DES EXERCICES**

# **I- INTRODUCTION AU LANGAGE PASCAL :**

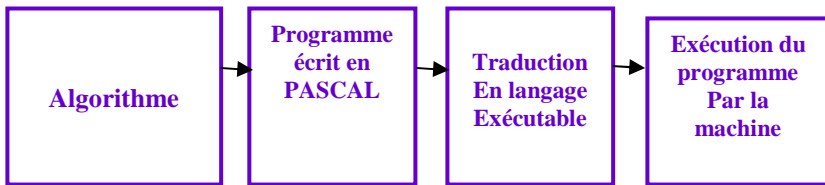
## **1- Caractéristiques générales du langage pascal :**

### **1.1 Généralités :**

Un algorithme est le résultat d'une démonstration logique de résolution d'un programme. La méthode de résolution doit alors être transcrite en une suite ordonnée d'équations, de définitions et d'actions, qui forment le corps de l'algorithme.

Pour concrétiser réellement un algorithme, il faut au préalable le traduire en un programme d'un langage de programmation compréhensible par l'ordinateur.

Pour cela, le langage PASCAL est un langage de programmation très structuré qui respecte la forme algorithmique, son écriture est alors aisée.



**Cheminement d'un programme**

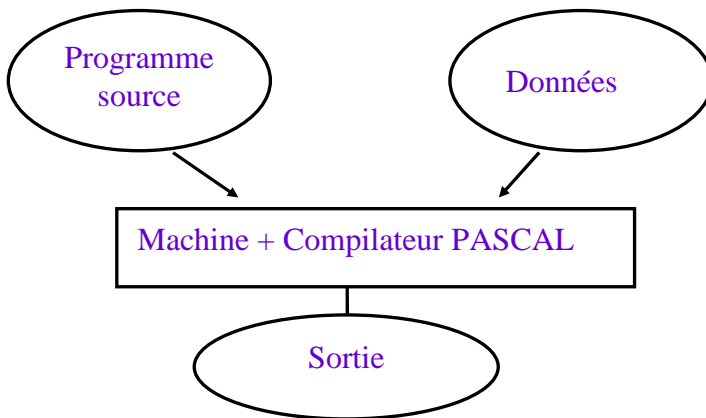
Le langage de programmation PASCAL a été conçu et développé au début des années 70 par NIKLAUS WIRTH à la " EIDGENOSSE TECHNISCHE HOCHSCHULE DE ZURICH " (Ecole polytechnique de Zurich). Jusqu'à présent PASCAL a été largement utilisé dans les universités, les écoles d'Ingénieurs et les centres de recherches en Informatique, mais l'apparition de compilateurs PASCAL (Programme qui permet de traduire un programme écrit en PASCAL en un programme compréhensible par la machine pour y être exécuter) sur les micro-ordinateurs a conduit à un développement considérable de ce langage.

Actuellement, il existe plusieurs environnements PASCAL parmi les plus importants on cite MS PASCAL de Microsoft et TURBO PASCAL de BORLAND.

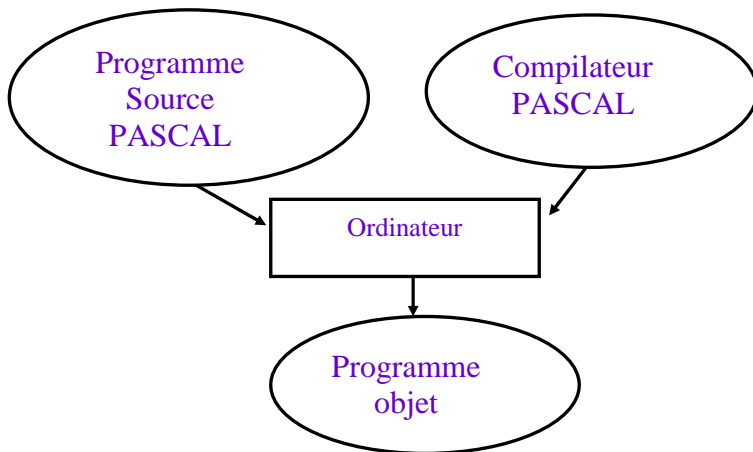
## 2- Environnement technique du langage pascal :

Encore une fois, le langage PASCAL est un langage structuré possédant une clarté dans la disposition des instructions, par conséquent, il s'avère le langage d'enseignement par excellence.

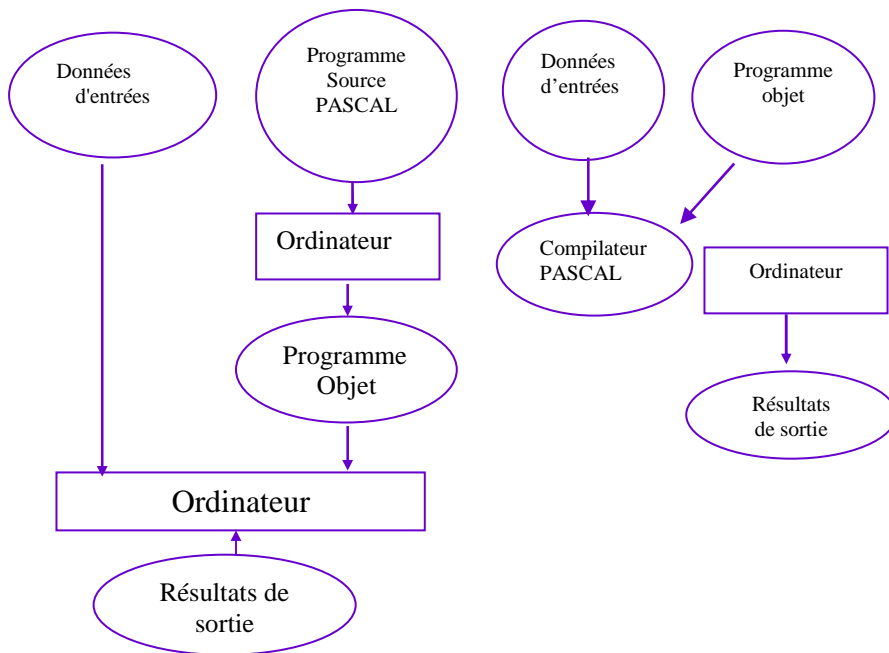
D'une manière globale pour exécuter un programme écrit en PASCAL (programme source) dans un ordinateur, il faut au préalable le traduire en un programme compréhensible par la machine (ordinateur).



Autrement, on peut considérer dans un premier temps que le programme est le COMPILATEUR PASCAL et la donnée est le programme source. Le résultat sera la traduction de notre programme source (en PASCAL) en un programme formé d'instructions machine : LE PROGRAMME OBJET.



Ensuite, l'ordinateur est alors utilisé à nouveau : son programme sera le programme objet et ses données d'entrée nos propres données, produisant ainsi le résultat attendu.



## SCHÉMA RÉCAPITULATIF

Dans les deux étapes, il faut prévenir des erreurs à savoir :

- L'erreur de compilation détectée lors de la traduction :  
Si par exemple nous écrivons Begin mais que nous omettons le end. Le compilateur détecte et signale l'erreur.
- L'erreur d'exécution est détectée pendant le déroulement du programme objet : Si au programme Racine Carrée suivant nous donnons la valeur -1 à x, le programme sera mis en défaut parce que SQRT n'accepte pas d'argument négatif.

PROGRAM Racine Carrée (input, output);

VAR x: Real;

BEGIN

Read (x);

Write (sqrt(x));

END.




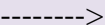

### 3- Notation fondamentales (syntaxe) :

#### 3.1 Le Diagramme de Syntaxe :

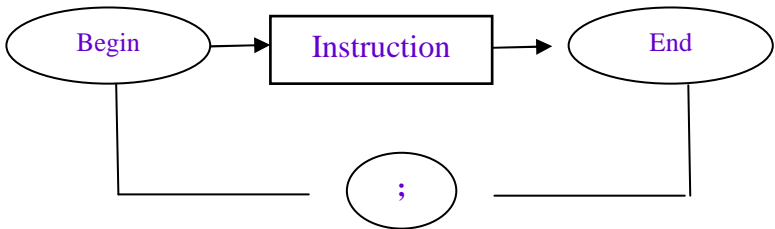
Tout programme PASCAL est décrit par sa syntaxe et sa sémantique. La syntaxe établit les règles d'écriture que l'on doit utiliser pour formuler chacune des instructions, tandis que la sémantique indique comment il faut enchaîner instructions et actions pour exécuter une application.


Le graphisme représentant les règles d'écriture et leur enchaînement est appelé un diagramme de syntaxe.

Les symboles utilisés :

	Contient un mot dont la signification est réservée au langage (il est appelé Mot-clé)
	Indique des ensembles de déclarations ou de traitements
	Contient un caractère ou un symbole ayant une signification particulière
	Donne le sens de parcours du diagramme de syntaxe
	Signifie que plusieurs chemins sont possibles à partir du nœud.

**Exemple :** soit le diagramme



<b>BEGIN</b>	<b>Indique le début du programme</b>
<b>INSTRUCTION</b>	Désigne l'un des éléments du traitement ; ce que peut être : <ul style="list-style-type: none"><li>- Une Instruction d'écriture</li><li>- Une Instruction de lecture</li><li>- Une Instruction d'exécution de test</li></ul>
<b>;</b>	Caractère séparateur d'instructions
	Indique le sens de parcours du diagramme .Dans cet exemple, s'il existe encore une instruction à réaliser le cheminement se fera vers le caractère ; sinon vers END.
<b>END.</b>	Indique la fin du traitement.

## II- ÉLÉMENTS DE BASE :

### 1- Alphabet pascal :

La représentation écrite du langage est construite à l'aide d'un jeu de caractères ou symboles de base composé de lettres, de chiffres et de caractères spéciaux assurant une fonction bien précise, ainsi que du caractère Espace.

Ce jeu est constitué des caractères suivants :

- Les 26 lettres de l'alphabet : A à Z, a à z.
- Les 10 chiffres : 0 à 9.

- Les caractères spéciaux suivants : + - \* / = ^ < > ( ) [ ] { } . , : ; ' # \$ \_

### 2- Élément d'un texte pascal :

#### 2.1- Les mots clés pascal :

Ce sont des mots reconnus par le compilateur, ils constituent le vocabulaire de base du langage PASCAL. Ils servent à définir le rôle des instructions ou les caractéristiques des données.

#### Exemple :

FUNCTION, INTEGER, TYPE sont des mots clés.

Certains mots clés sont réservés à un usage bien précis et ne peuvent être utilisés hors de leur contexte. Ce sont les mots réservés que le programme ne devra employer que dans leur rôle prédéfini. Ce sont :

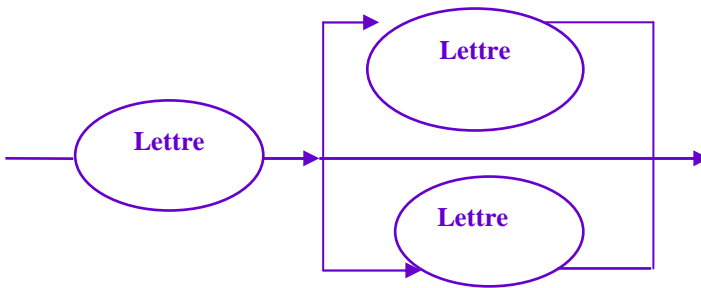
AND	ARRAY	BEGIN	CASE	CONST
DIV	DO	DOWNT	ELSE	END
FILE	FOR	FORWARD	FUNCTION	GOTO
IF	IN	LABEL	MOD	NIL
NOT	OF	OR	PACKE	PROCED
PROGRAM				URE
TO	RECORD	REPEAT	SET	THEN
	TYPE	UNTIL	VAR	WHILE.

## 2.2- Les identificateurs :

Les objets manipulés sont nommés par les identificateurs qui sont choisis par le programmeur. Ils désignent des noms de variables, d'ensembles de constantes figuratives, de procédures, ...

Un identificateur est une suite de lettres et de chiffres le premier symbole étant obligatoirement une lettre.

La syntaxe des identificateurs est donnée par le diagramme suivant :



### Exemple :

Exemples Valides	Exemples non Valides
<b>NOM</b> <b>PRENOM</b>  <b>TURBOPASCAL</b> <b>F22</b> <b>MINUSCULE</b> <b>EXO</b>	9 NOM TURBO PASCAL END. CH 22 : Minuscule

**REMARQUE :** Il est conseillé de déclarer les identificateurs sur une longueur de 8 caractères pour respecter les normes standards et pour éviter d'avoir des noms trop longs dont l'écriture pourrait devenir très fastidieuse.



### 2.3- Les séparateurs :

Ils constituent la ponctuation du langage :

.. → Sépare les bornes d'un intervalle ;

, → Sépare les éléments d'une liste ;

: = → Symbole d'affectation ;

' → Encadre les constantes de type caractère et chaîne (il est dit **cote**);

. → Marque le point final d'un programme ;

; → Sépare deux lignes du programme ;

= → Précède la valeur dans une déclaration de constante ;

{ → Début de commentaire ;

} → Fin de commentaire ;

[ et ] → Encadrent la liste d'indice d'une variable indicée ;

→ Sépare 2 mots consécutifs (il s'agit de **l'espace**) ;

(et) → Encadrent les listes associées à des mots clés ;

### 2.4- Les opérateurs :

Ils indiquent un traitement à l'ordinateur. On trouve les opérateurs classiques :

#### • Opérateurs arithmétiques :

- Addition : +
- Soustraction : -
- Multiplication : \*
- Division : /

#### • Opérateurs de comparaison :

Toutes les relations d'ordre :

- Inférieure ou Egal : <=
- Supérieur : >
- Egal : = etc. ...

Ils sont généralement utilisés pour exprimer une condition:

**Exemple** : SI  $A \geq 10$  alors ... ( en Algorithmique)  
IF  $A \geq 10$  THEN.... ( en Pascal)

### • **Opérateurs Logiques** :

En PASCAL, ils sont désignés par des mots clés :

**AND** : ET logique

**OR** : OU logique

**NOT** : Négation logique etc...

Ils permettent de combiner plusieurs conditions :

**EXEMPLE:**  $(S > 10)$  ET  $(B \leq 10)$   
 $(S > 10)$  **AND**  $(B \leq 10)$

### • **Opérateurs d'ensembles** :

\* : Intersection d'ensembles ;

+ : Union d'ensembles ;

### • **Opérateur d'appartenance** :

Il est noté par : **IN**

## 3- **Les commentaires** :

Les commentaires, sont des suites de caractères encadrées par les symboles : { et } et qui servent pour argumenter les programmes (décrire se que fait une instruction par exemple).

**Exemple** : { test de programme }

**Remarque** : Il est très important de mettre des commentaires dans les programmes afin de les rendre plus lisibles et compréhensibles.

#### 4- Structure générale d'un programme pascal :

Un programme est une suite ordonnée d'instructions qui permettent la réalisation d'un traitement donné exécutable par l'ordinateur.

La structure générale d'un programme PASCAL est composée de 3 parties :

- 1- Une identification du programme (en- tête) ;
- 2- Une partie déclaration ;
- 3- Une partie instructions toujours terminée par un point.

<b>IDENTIFICATION</b>	L'identification du programme est assurée par un en-tête qui spécifie le nom du programme suivi d'une liste de paramètres de l'environnement.
<b>DECLARATIONS</b>	La partie déclarations décrit les objets sur lesquels porteront les traitements. Ils sont nécessairement réservés avant tout traitement les affectant dans la partie INSTRUCTIONS.
<b>INSTRUCTIONS</b>	C'est la partie principale du programme elle décrit l'ensemble des opérations à effectuer. Elle commence toujours par le mot-clé BEGIN et ce termine par le mot END suivi d'un point.

## Partie IDENTIFICATION

```
PROGRAM ( prog 1 ( INPUT , OUTPUT ) ;  
          { * partie déclarations * } ;  
BEGIN  
          { * partie instructions * } ;  
END.
```

Fin du programme

Ce programme a le nom : prog1, ses paramètres d'environnement sont **INPUT** et **OUTPUT** (ce sont des fichiers standards d'entrée de données et de sorties de résultats) et est constitué d'une partie déclarations et d'une partie instructions.

**PROGRAM** : Le début du programme PASCAL est défini par le mot réservé 'PROGRAM' suivi du nom du programme attribué par l'utilisateur, des paramètres entre parenthèses et du caractère point-virgule (;) indiquant la fin de la ligne.

### EXEMPLE :

Nom du programme      Séparateur de paramètres

**DECLARATIONS** : La partie DECLARATIONS quant à elle, est structurée en cinq (05) parties distinctes remplissant chacune une fonction précise:

**LABEL** ..... Déclaration d'étiquettes

**CONST** ..... Déclaration de constantes

**TYPE** ..... Déclaration des types de données

**VAR** ..... Déclaration des variables

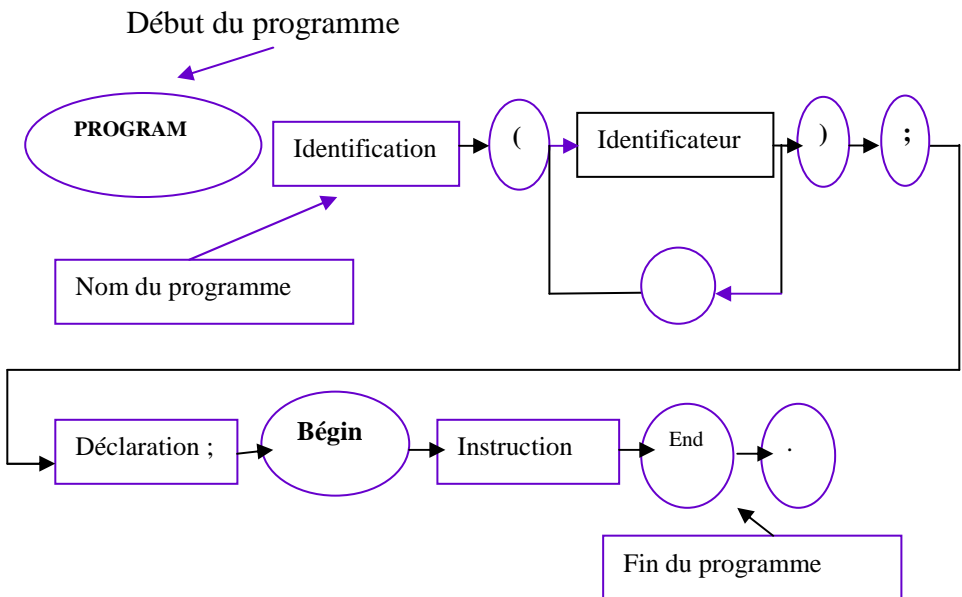
**PROCEDURE/FUNCTION** ... Déclaration de sous programmes

**Instructions :** La partie INSTRUCTIONS est constituée d'une suite ordonnée d'opérations élémentaires à effectuer. Elle commence et se termine respectivement par les mots réservés **BEGIN** et **END**, ce dernier est toujours suivi d'un point

**PROGRAM** Example (INPUT, OUT PUT);

**REMARQUE :** Les déclarations et les instructions sont séparées par des points virgules (;).

Afin d'illustrer l'utilisation des diagrammes de syntaxe la structure générale d'un programme PASCAL peut être représentée de la manière suivante :



### **DIAGRAMME DE SYNTAXE DE LA STRUCTURE GÉNÉRALE**

## RÉSUMÉ :

Dans cette leçon nous avons appris que le langage Pascal est un langage de programmation évolué, doté d'un compilateur qui traduit tout programme source écrit en Pascal en un programme machine exécutable.

Pour écrire un programme en Pascal il faut respecter sa structure qui est composée des trois parties suivantes :

- 1- L'identification ou l'en-tête du programme qui consiste à lui associer un nom et lui définir quelques paramètres d'environnement.
- 2- Les déclarations pour décrire tout les objets que manipule le programme tel que les constantes, les variables, les types ...etc.
- 3- Les instructions qui jouent le rôle principal dans le programme, se sont des opérations de base tel que la lecture, l'écriture, les opérations mathématiques...etc.

La partie instruction doit commencer par le mot réservé BEGIN et se terminer par le mot réservé END suivi d'un point.

## **EXERCICES D'APPLICATION :**

### **EXERCICE N° 01 :**

Les identificateurs suivants sont-ils correctes ? Justifiez votre réponse.

<b>IDENTIFICATEUR</b>	<b>REPONSE</b>
Essai	
10B	
A*	
VA R	
CASE	

### **EXERCICE N°02 :**

Soit le programme suivant qui affiche un simple message :

```
PROGRAM 1Message (INPUT, OUTPUT);
```

```
BEGIN
```

```
WRITE ('Bonjours tout le monde');
```

```
END;
```

Donner les différentes erreurs de syntaxe.

### **EXERCICE N°03 :**

Soit le programme suivant qui calcule la somme des carrés de deux nombres puis l'affiche à l'écran. Discutez sa syntaxe.

```
PROGRAMME Scarre (INPUT, OUTPUT),
```

```
VAR
```

```
    A, B, C : INTEGER
```

```
BEGIN
```

```
    READ (A, B);
```

```
    A = A*A;
```

```
    B = B * B;
```

```
    C = A + B;
```

```
    WRITE (C);
```

```
END.
```

## CORRIGÉ DES EXERCICES :

### EXERCICE N°01 :

IDENTIFICATEUR	RÉPONSE
Essai	Correcte
10B	Incorrecte Un identificateur doit commencer par une lettre
A*	Incorrecte Un identificateur ne contient pas des caractères spéciaux
VA R	Incorrecte Un identificateur ne contient pas le caractère espace
CASE	Incorrecte Il s'agit d'un mot clés

### EXERCICE N°02 :

Il existe deux erreurs dans le programme :

**Première :** L'identificateur du programme '1message' doit commencer par une lettre ;

**Deuxième :** La fin du programme c'est END. Au lieu de END ;

### EXERCICE N° 03 :

On remarque qu'il existe trois erreurs de syntaxe :

**Première :** Le mot clés PROGRAM sans E ;

**Deuxième :** L'en-tête du programme se termine par point virgule au lieu de virgule ;

**Troisième :** En fin des déclarations on aura le point virgule (parés INTEGER).